Listing of Claims:

1. (currently amended) A computer-implemented, incremental process for executing an application servo in a client device based on a specified set of matching criteria, the process comprising the steps of:

selecting a servo to provide services;

identifying a datasource associated with the selected servo;

initializing an execution context tree structure by creating a root node of the context tree associated with an initial instruction of the servo;

choosing a context of the context tree that satisfies the matching criteria;

executing an instruction of the servo associated with the chosen context;

responsive to said executing step, creating zero or more new child contexts in the context tree, each new child context including content defining a current internal evaluation state of the process; ~~and~~

repeating said choosing, executing and creating steps over subsequent instructions of the servo until no context satisfies the matching criteria;

responsive to changes to the datasource, marking dependent contexts as unverified;

choosing a marked context of the context tree;

performing an instruction of the servo associated with the chosen context;

responsive to said executing step, creating zero or more new child contexts in the context tree and removing or modifying zero or more existing child contexts;

unmarking the chosen context and marking zero or more dependent contexts as unverified; and

repeating said choosing, performing, creating, removing, modifying, unmarking and marking steps over subsequent instructions of the servo until no contexts are left marked.

2. (original) A process according to claim 1 wherein the content of the child context includes:

a pointer to an element within the selected servo; and

a pointer that identifies a current data context by pointing into a source tree.

3. (original) A process according to claim 1 wherein the content of the child context includes:

a reference to a parent context;

an ordered, potentially sparse, list of pointers to zero or more child contexts; and definitions for any symbols introduced by the context.

4. (currently amended) A process according to claim 1 further including, responsive to said executing and performing steps, creating zero or more child spacers in the context tree representing unmaterialized child contexts; and wherein said choosing a context includes choosing either a context or a spacer.

5. (original) A process according to claim 4 wherein the context tree is implemented using a relative b-tree structure, and each spacer is reflected in an interior node entry in the relative b-tree structure to facilitate searching unmaterialized contexts.

6. (original) A process according to claim 1 wherein the b-tree node entry includes a field to track a linear value associated with a graphical display output object.

7. (original) A process according to claim 1 wherein the process creates and maintains both the context tree and a geometry tree, the geometry tree representing the spatial structure of a predetermined graphical user interface.

8. (original) A process according to claim 1 wherein the servo is defined using a servo definition language that references XML schema definitions as its core vocabulary.

9. (original) A process according to claim 8 wherein the servo definition language comprises:

application data schema;

transformation rules; and

opportunity rules.

10. (currently amended) An interpreter stored in a computer-readable medium, the interpreter for interpreting a servo definition language for defining a distributed application that supports disconnected operation, the language comprising the following types of rules:

application data schema;

transformation rules;

transaction handling rules; ~~and~~

interface object specifications;

opportunity rules to realize automatic extension or integration of servos through opportunity-based linking of an interface component representing an instance of a schema fragment to a template.

11. (currently amended) An interpreter ~~servo definition language~~ according to claim 10 further comprising access rules.

12. (canceled).

13. (currently amended) An interpreter ~~servo definition language~~ according to claim 10 ~~12~~ wherein the template specifies at least one of a transformation rule, a transaction handling rule and an interface object specification.

14. (currently amended) An interpreter ~~servo definition language~~ according to claim 10 and further comprising an abstract interface object definition.

15. (currently amended) An interpreter ~~servo definition language~~ according to claim 10 wherein the application data schema comprises an XML-based schema.

16. (currently amended) An interpreter ~~servo definition language~~ according to claim 10 defined using XML schema definitions XSD as the core vocabulary.

17. (currently amended) An interpreter ~~servo definition language~~ according to claim 10 including a view element for selecting a group of the said transformation rules to define at least a part of an output interface.

18. (currently amended) An interpreter ~~servo definition language~~ according to claim 10 including a storage declaration element that enables an author to reserve and name persistent storage for use by the servo and any other servos authorized to access the corresponding data.

19. (currently amended) An interpreter ~~servo definition language~~ according to claim 18 wherein the storage declaration element includes a locally scoped name for a corresponding storage tree and identifies a schema to which the storage tree must conform.

20-40. (canceled).